Document Assembly in Civil Legal Services

An overview of concepts, issues, tools, and methods

Marc Lauritsen
Equal Justice Conference
Cleveland – April 2002

This paper is largely a selection of highlights from "Web-based document assembly in the civil legal assistance community," a white paper prepared by Capstone Practice Systems in November 2001 as part of a project supported by The Legal Services Corporation, Legal Services of Indiana, and ProBono.net. The full version of that paper and related materials are available at http://www.lstech.org/workgroups/doc_assembly.

Contents

Introduction

Technologies for the automated production of legal documents have been in use for well over twenty years, and increasingly sophisticated applications can be found on law office desktops across the profession. Document assembly tools offer great improvements in both productivity and quality for the delivery of legal services. But for various cultural, political, and economic reasons, actual use remains limited to discrete islands of enthusiasts. The nonprofit legal services world in particular has made relatively little use of this technology.

In the last several years these applications have begun to be deployed over the Web, promising dramatically greater scope and easier distribution. Again, take-up has mostly been by experimenters and early adopters. But the opportunities have been recognized by many programs, and funds are now becoming available to pursue them on a large scale. This is an important time for coordinating efforts and leveraging common resources.

"Document assembly" is used here in a broad sense, covering

- both the automated production of documents and the "intelligent interviewing" that typically precedes it;
- both word processing documents and official or "graphical" forms;
- applications used by lawyers and other advocates, as well as those used by *pro se* litigants and other individuals doing legal work on their own behalf;
- both fully online configurations and mixed topologies, e.g. involving desktop applications and local area networks.

Using computers to assist people in the preparation of law-related documents can be accomplished in a wide variety of ways, involving a bewildering array of technical and managerial choices.

WHAT is document assembly?

Basic concepts

Document assembly. Computer-aided drafting. Document modeling. Document automation. There are many names for software tools that help people quickly generate certain types of well-structured documents. Contracts and wills are good examples. A lawyer, paralegal, secretary, or do-it-yourselfer responds to a series of dialogs and prompts, often from within a familiar word processing program, and the system assembles a draft document. Or the user picks forms, clauses, and other document components as needed from libraries of alternatives.

Sometimes, an organization develops a custom system with one of the document assembly "engines" mentioned below, using its own forms and experience. This can require a fair amount of up-front time and tedious work (thinking through and

programming many possible alternatives) but can result in excellent leveraging of practical legal knowledge.

Other times, the document assembly system is one that is obtained from a legal publisher or document assembly vendor, designed to produce specific types of documents valid in certain jurisdictions. Some well-known off-the-shelf systems include Immigrant Pro from Immigrant Software (http://www.immigrantsoftware.com), WillMaker and LeaseWriter from Nolo Press (http://www.nolo.com), and JC Forms from Capsoft (http://www.capsoft.com), which generates approved California court forms. TurboTax from Intuit (http://www.quicken.com/taxes/) is probably the most popular program of all time with form automation features. Sometimes legal services organizations develop their own systems and make them available to fellow organizations. One example of such sharing is the Greater Boston Legal Services family law system, which is being used by a variety of organizations in Massachusetts.

Either way, the basic goal is to capture some of the regularities underlying the documents—what sections, paragraphs, sentences, and words go where under what circumstances. The document assembly engine provides a kind of power steering for lawyers and others to make choices and specify details like names, numbers, dates, and phrases. Instead of cutting and pasting, you can pick desired options or alternatives from lists; instead of searching and replacing phrases like "Plaintiff name" with your client's name, you can respond to questions and let the computer do the clerical work.

While terminology varies among programs, there is usually a "template" that represents a model of particular kind of document, with "variables" placed in locations that change from case to case. When the template is run, the user answers questions corresponding to the variables (posed in a series of interview-style dialogs), the answers are stored in some kind of "answer file," and the desired document is generated. Typically a given answer file stores all the data relevant to a single client or client-matter, and that answer file can be used to generate more than one document or form (e.g. a complaint for divorce, a financial statement, and various motions in a family law system). Answers can be changed later (e.g. a correction to the name of the client) and the document(s) can be re-generated. The generated document is usually in some textual format (e.g. Word, WordPerfect, RTF) and can be freely edited after assembly.

In addition to basic point-and-shoot clause selection and fill-in-the-blanks variable replacement, these systems can store drafting rules and other kinds of practitioner knowledge that can be used to guide the hand of novices and experts alike. For example, a divorce system can be designed to ask the user about the client's state of residence, marital status, financial situation, and number of children and, based on the answers and follow-up questions, insert appropriate clauses into the complaint for divorce and associated motions. Document assembly technology has been applied to everything from simple thank-you letters to elaborate expert systems that advise on the laws of many jurisdictions and generate document sets that can reach into hundreds of pages.

Web-enabled document assembly

The World Wide Web opens up several new opportunities for organizing and delivering document assembly applications. Any or all of the major components – engine, templates, answers, documents, help material – can be served from or stored on a Web server, providing location independence, multi-user access, ease of use, and other benefits characteristic of the Web. For advocates, a big advantage of Web-based implementations is the centralization and instant updating of template collections. For *pro se* users, they allow access to robust document automation without requiring special purpose local software to be purchased, installed, configured, and maintained. Often just a browser is required, together with an Internet connection and a printer – tools that are available in most public libraries. For IT professionals (and budget conscious managers) a single centralized server and staff can economically provide document assembly capabilities to hundreds or thousands of users.

Web-based document assembly environments can not only replicate much of the functionality we've seen on the desktop, but add interesting new features, such as hyperlinks in the assembled documents that take you back to associated questions, where you can enter or change answers and reassemble the document. And, helpful links to ancillary websites can easily be placed in the browser-based dialogs.

There are at least two different forms of "Web-based" document assembly. In the first case, sometimes referred to as a "fat client" scenario, the templates are accessed and downloaded from a Web site, but run in a regular local computer application, like Rapidocs Classic or HotDocs. In the second case, a "thin client" computer is all that is needed, and the assembly process either happens entirely on the Web server, or via a small application that serves as a "plug-in" to the local browser, such as an ActiveX control, used by Rapidocs, or a Java applet, used with Grantha. Several vendors support both methods.

Possible disadvantages of Web-based approaches include disrupted access due to server downtime, some loss of functionality (due to the limits of the browser interface), the necessity of an internet connection, and some greater difficulty in linking to databases, case management systems, and other third party applications that may run locally in legal services offices.

Important distinctions

Some recurring distinctions are important to make:

1. *Word processing documents vs. graphical forms.* Document assembly generally encompasses both freely editable word processing documents and fixed-format, "graphical" forms, where the background is static and information typically can only be placed in pre-designated fields. The terminology for these two kinds of documents varies, and can be a source of confusion. But most contexts we care about require both kinds of documents.

2. *Questioning and advice vs. document generation.* One of the characteristics of most document assembly applications is that users provide information and make drafting decisions through questionnaire-like screen dialogs that are *outside* of a target document. The document being generated may be visible during the interview, and the user may be able to access it to revise answers or edit passages, but usually there is a discrete interface in which questions can be asked and advice given. Many document assembly tools can in fact be used to produce information gathering modules, advisory systems, and intelligent checklists that needn't result in any traditional document at all.

3. *Advocates vs. self-help users.* Document assembly technology can be and is being used both by advocates providing services for clients and by individuals doing work for themselves. Software selection criteria and project planning can differ dramatically for the two target communities. And even among advocates, there can be important differences between the needs of staff advocates and pro bono or other volunteers. This report tries to cover most issues in a common fashion, pointing out differences when appropriate. There are also hybrid *pro se*/advocate scenarios, in which the client answers an online questionnaire on his or her own, either in the law office or elsewhere, and the answer file goes to the attorney for further review, revisions, and actual document drafting.

4. *Users vs. developers.* Document assembly software typically involves distinct tools and interfaces for "end users" and developers. There are many features and issues that can be critical for people charged with developing applications that are irrelevant to the ultimate users, and vice versa. Some software choices offer excellent end user interfaces but clumsy development tools, and vice versa.

5. *Accessing templates on the Web for local processing versus assembling them as part of an online session.* See previous section.

These distinctions often are combined. For instance, a given online document assembly initiative might involve interactive questioning and advice that is entirely browser-based, but document generation that happens on the desktop. Or word processing documents that are assembled on the server, but graphical forms that are built locally. Or one approach that is followed for self-help users and another for advocates. In characterizing any such implementation, you really need to ask *what* is happening *where*, *when*, and *how* for *whom*?

WHY deploy document assembly?

Benefits for advocates

Document assembly systems can provide the following benefits from the advocate's perspective:

- **Quality assurance.**
  - **Correctness**. Assure that client information is correct on all forms; assure that forms are filled out in the correct manner (e.g. that income is listed in weekly, monthly or annual amounts as appropriate on a court form).
  - **Completeness**. Assure, for example, that all appropriate requests for relief are made (or at least considered by the advocate).
  - **Consistency**. Assure that the latest language, for example, on interrogatory questions, is used by all advocates.
  - **Standardization**. Encourage legal services organizations to reconcile language differences among models used by various advocates and standardize on best practices.
- **Productivity/efficiency**. Dramatically reduce the time required to draft complex documents such as financial statements in divorce matters; enable paralegals and students to create first drafts of documents previously done by attorneys; enable secretaries to create first drafts that were previously done by paralegals; enable legal services programs to serve more clients with the same or fewer resources.
- **Responsiveness**. Dramatically reduce total elapsed time between client interviews and court filings, or the settlement of a case and its documentation.
- **Process improvement**. Allow instant sharing (across time, staff, and offices) of client data captured and stored in answer files; facilitate access and re-use of client information already stored in a case management system. Better understand and re-engineer these kinds of processes through the very work of automating them.
- **Training and continuing education**. Guide less experienced advocates through the correct questions and options via dialog screens; provide optional help screens, with explanatory text, to teach legal and advocacy skills.
- **Consolidation of expertise**. Capture the substantive knowledge of more experienced and specialized attorneys (e.g. the right questions, options, language, strategies) so that:
  - Knowledge is shared within an office and across legal services offices.
  - Knowledge is preserved in case of staff turnover.
- **Job satisfaction and enrichment**. Liberate advocates to focus on more challenging and satisfying tasks (e.g. the stuff attorneys went to law school for).
- **Access to legal services**. Help discharge professional responsibility to improve access to legal services through better technologies both for self-help and assisted scenarios..

Benefits for *pro se* litigants and other self-helpers

From the perspective of someone pursuing legal work on their own behalf, a document assembly application can provide:

- Access to information and assistance when other sources are unavailable or ineffective.
- Meaningful guidance at a meaningful point in the process.
- Ability to work at their own schedule and pace.
- Generation of forms that comply with the format and content requirements of a court or agency.

Effectively implemented, Web-based document assembly technology has the potential to produce *transformative* increases in access across the spectrum of non-advocate users. The return on investment of self-help-oriented initiatives can be quite spectacular, even compared to enhanced tools for advocates.

Obstacles and opportunities

Document assembly technology is mature and ready. It offers dramatic benefits in efficiency, quality, and job satisfaction. It can usually work in your existing computing environments. It can be a powerful supplement to case management, electronic filing, and *pro se* initiatives.

So why doesn't the civil legal assistance community make much use of document assembly? How do we get from pilots and demonstrations to mainstream and routine implementations? This is not only a matter of good technology and effective project management, but of courage, leadership, and cultural sensitivity. Our problem has not been lack of ideas or tools. There are deep cultural and managerial challenges that need to be overcome. Here are just a few of the factors often identified:

- Unawareness of available tools and how to tap their benefits
- Lack of in-house resources and expertise for automating templates
- Personal and organizational inertia
- Innovation overload
- Conceptual difficulty
- "It's nobody's job"
- Professional arrogance. Many lawyers feel that their specialized knowledge is not amenable to any useful forms of automation.
- "Not invented here." It is often important to standardize on document models before automating. Standardizing can be good (it forces advocates to think about why they have different models and chose the best one or two) but it can be a huge obstacle too (attorneys need precious time to focus on this and of course, everyone thinks his/her language is better).
- Fear of lost creativity. People worry that form systems can produce routinized, unimaginative, compromised practice. While this is a legitimate concern, it should also be pointed out that by automating the routine and mechanical aspects

of practice, people will have more time and energy to attend to the humanistic and creative dimensions.
- Too busy bailing … (to repair the hole in the boat)

On the other hand, our challenge is not just to discover and apply established technologies and well-understood methods – there are fascinating questions about how to use this technology to best effect in support of the equal justice agenda.

HOW do you run one of these projects?

Some common issues

A legal services organization considering a Web-based document assembly project needs to consider issues like the following:

1. Document assembly technology for *whom*? - LSC grantee staff attorneys? or also paralegals, secretaries, students? staff in non-LSC funded legal aid offices? bar association initiatives? pro bono? law school clinics?

2. Document assembly for advocates only, *pro se* litigants and the general public, or both?

3. Pure Web-based, or also desktop applications?

4. Basic document generation, or also associated functions like document management, workflow, knowledge management, and case management integration?

5. How far would you like to go in acquiring or building substantive templates, as opposed to the just the "plumbing" (e.g. a Web server and some enabling software) and some basic forms?

6. What general approach will you take to staffing the development, maintenance, and support functions?

7. To what extent will the applications be distributed among desktops and local servers, and to what extent centralized in pooled resources like a shared server farm?

What's involved? The major ingredients

To provide the benefits of automated document drafting to a community of users, you need the following basic ingredients:

1. A delivery environment (hardware, networks, and general software)
2. An underlying "engine" (e.g., Adobe Acrobat, GhostFill, Grantha, HotDocs, Rapidocs, SmartWords, or custom equivalent)

3. Some “content” (intelligent templates)
4. Educated end users
5. Arrangements for building and maintaining document automation aspects of the delivery environment
6. Arrangements for maintenance of content
7. Arrangements for support of users

To the extent that you need or want some of #3 (content) to come from within your own community, you need

8. Educated template developers
9. Arrangements for support of developers

And to achieve #4 and #8, you need

10. Training of users and developers

Basic technology choices

There are two major categories of technical decisions to be made: (1) what software platform(s), existing or new, will you base your solution on?; and (2) what kind of “architecture” will you follow in allocating processes among various server and client machines?

On the first question, choices include:

1. going with an existing, specialized document assembly tool
2. building a custom solution using more generic software
3. combining the above two strategies

The three basic architectural options are:

1. Traditional desktop / local or wide area network (“fat client”)
2. Serving applications from Web site, but running them with local software (ditto)
3. Pure server-based (“thin-client”)

Specialized engines

Computer-aided document drafting can be accomplished through any number of software tools. Macro and merge features built into today's word processors are of course often used. Similar features are also available in some database programs, spreadsheets, groupware applications (like Lotus Notes), and general purpose programming tools like Visual Basic. But quite a few specialized programs have emerged for building legal document assembly applications—variously dubbed “engines,” “platforms,” “authoring environments,” and the like. And there are distinct products and vendors for other vertical markets like accounting, banking, health-care, and insurance, with little apparent cross-pollination, despite great functional similarity.

There is a full list of law-oriented document assembly engines below. Several of these were the focus of workshop held in New York City in February as part of the project referred to on the title page of this paper, and detailed information about them is available at http://www.lstech.org/WorkGroups/DA/index.htm.

Some other platforms and approaches

Adobe Acrobat (PDF)

A number of online document assembly sites use Adobe Acrobat (http://www.adobe.com) technology to fill graphical forms. Out of the box, in desktop mode, Acrobat offers extensive functionality for on-screen filling of forms. Adobe also distributes a free "Reader," which allows the user to view and fill documents, including forms, created by the full version of Acrobat, but not to save the filled forms, or the associated answers. Acrobat files use a proprietary format, referred to as "PDF," these letters forming the file name extension. Because the Reader is free, the PDF format is a popular choice among government agencies and court systems when it comes to distributing forms in electronic versions. Many electronic filing systems require it.

Advantages of an Adobe Acrobat approach
➢ Acrobat provides reliable display and printing across a wide range of platforms. ➢ The Acrobat reader program is free, widely available, and often already present on a client computer. No additional local software is required. ➢ Adobe has a corporate giving program that can make the authoring software available free. ➢ Acrobat forms can be pure facsimiles of court and governmental forms that can't accidentally be changed. ➢ Help notes and nonprinting instructions can be made available. ➢ Version 5.0 supports collaborative editing and annotation of forms from within a Web page. ➢ The forms are easy to fill-out and can do basic calculations. ➢ Security can be imposed to prevent forms from being changed, copied from, or even printed.
Disadvantages of an Acrobat approach
➢ There is no built-in support for a separate "interview" that can guide a user through a complex form. ➢ Users can't save filled forms or associated answers with the free Reader. ➢ Documents that require post-assembly editing in a word processing environment cannot easily be supported. ➢ Acrobat forms lack some features of more sophisticated programs like HotDocs Automator, whose forms can include interactive dialogs, conditional fields, conditional and repeated pages, more sophisticated calculations and error checking, built-in support for answer storage between sessions and use across forms, and advanced overflow handling (shrink to fit, automatic appendix creation.)

Jnana

Jnana, from Jnana Technologies Corporation – http://www.jnana.com – is a pure Web-based "inferencing engine" that can be used to design and deliver interactive sessions that guide a user through facts and considerations to reach a legal (or other) decision, based on a set of pre-programmed rules. It is being used in several projects at the Public Interest Clearinghouse in California. Jnana does not have a document assembly module per se, although it can create texts as part of a session, and now has a built-in link to HotDocs Online and could be similarly integrated with other platforms.

Custom software

Of course, one final strategy is to proceed without any specialized document assembly engine at all. Many law-related Web sites have used generic Web technology (active server pages, Cold Fusion, etc.) to deliver interactive question-and-answer sessions and build customized documents.

While all document assembly projects involve a substantial amount of custom work, most knowledgeable document automation developers recommend not "rolling your own" *tools* if you can avoid it. Especially for advocate-oriented implementations, the vast number of specialized document automation features found in commercial tools are very hard to replicate. See the following table for a summary of some pros and cons of commercial and custom tools.

Advantages of custom development over commercial tools
➢ Flexibility in features and interface ➢ May provide functions not available commercially ➢ Can be shared freely and collectively elaborated in an "open source" spirit
Advantages of commercial tools over custom development
➢ Usually much less expensive for comparable functionality, since costs are spread over a wide groups of users ➢ Less likely to be "orphaned" ➢ Access to a community of fellow users for support (technical and moral) ➢ Integration with third-party software may already be supported ➢ Up and running much more quickly

Selection criteria

Here is a high-level, five-part way of looking at the major considerations one should take into account in judging a technical solution for a Web-based document assembly project. The first two categories appropriately deal with the critical issue of **usability**. Note that some of these criteria only make sense for certain kinds of projects.

1. *User friendliness* – Ease of learning and use for the end user. Minimal local software requirements. Ability to run with any or no word processor. Speed of download and operation (of program components and templates). Ability to function offline. Ability to see document during the assembly process. Ability to observe during the assembly process how specific choices affect text in assembled documents. Ability to

support re-generation when there are post assembly edits. Ease of providing and accessing help.

2. *Developer friendliness* – Ease of learning and use for template authors. Functional completeness of development environment. Modularity: ability to re-use pieces of templates and associated programming within systems or across systems. Efficiency of development and maintenance. Top quality technical support for developers. Ability to document logic for lawyers and non-technologists. Flexibility, evolvability. Open interface. Support for graphical forms.

3. *Web readiness* – Deliverability through browser. Browser independence (ability to run at least in recent versions of Internet Explorer, Netscape Communicator, and AOL browsers). Interoperability and scalability of applications. Support for a variety of server operating systems and Web servers.

4. *Price worthiness* – Fair licensing prices and reasonable total cost of ownership.

5. *Vendor stability and partner-friendliness* – Financial and managerial stability. Extent and nature of existing user community. Willingness to partner and implement custom requirements, including support for different languages. Ability to establish and maintain long-term relationships.

Different solutions will be strong (and weak) in different areas, and their appropriateness will of course depend on the specific characteristics of the project a program is undertaking. It's useful to take a "balanced scorecard" approach to product comparison, and look for at least satisfactory ratings across all important dimensions.

WHERE can you get more information?

Articles and books

Introduction and overview

Lauritsen and Soudakoff, Power Tools for Document Preparation. *AmLaw Tech*, Spring 1998. Also at http://www.capstonepractice.com/amlaw6.pdf

Sprowl, Automating the Legal Reasoning Process: A Computer that uses Regulations and Statutes to Draft Legal Documents. 1 *Am. B. Found. Res. J.* 1-81 (1979)

Legal services applications

Lauritsen, Delivering Legal Services with Computer-based Practice Systems. 23 *Clearinghouse Review* 1532 (April 1990)

Product reviews

Soudakoff and Lauritsen, Shopper's Guide to Legal Document Assembly. *Law Office Computing*, October/November 1997 (Most of the article is available at http://www.docauto.com/locart.htm.)

Online services

Calkins and Granat, Client Self Help Strategies: Technology Educated And Assisted Pro Se With And Without Advocate Backup (1998). http://equaljustice.org/visions/TechConf/09-strategies.htm

Granat, From Legal Services to Information Services. Internet Practice Newsletter, May, 1997. Available at http://www.granat.com/legalservice.html.

Hornsby, William. Improving the Delivery of Affordable Legal Services Through the Internet: A Blueprint for the Shift to a Digital Paradigm. http://elawyering.org/what/improving.asp

Lauritsen, Assembling Documents on the Infobahn, *WORD Progress*, Summer 1997, p. 14, http://www.abanet.org/lpm2/newsletters/wp/su97laur.html

Project management

Lauritsen and Soudakoff. Unlocking the Power of Document Assembly. *Law Office Computing,* June/July 1999, p. 70-77

Artificial Intelligence

Branting, K., An Issue-Oriented Approach to Judicial Document Assembly, *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, pp. 228-235, ACM Press (1993)

K. Branting, C. Callaway, B. Mott and J. Lester, Integrating Discourse and Domain Knowledge for Document Drafting*, Proceedings of the Seventh International Conference on Artificial Intelligence & Law,* pp.72-81, ACM Press (1999)

Lauritsen, Knowing Documents. *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*. Amsterdam, June 1993.

Lauritsen, A Dispatch from the Document Automation Trenches. Workshop on Automated Document Drafting. *Seventh International Conference on Artificial Intelligence and Law*. Oslo, June 1999

Web sites

http://www.elawyering.org/
http://www.equaljustice.org/
http://www.lawofficecomputing.com

http://www.lawschoolconsortium.net/
http://www.technolawyer.com
http://www.unbundledlaw.org/
http://www.zorza.net/resources/lst-res.html

Document assembly development tools for legal contexts

(most not yet significantly Web-enabled)

ActiveDocs	http://www.keylogix.com
Boilerplate	http://www.wordsite.com/Boilerplate.html
CAPS	http://www.capsoft.com
DAS@H	www.das-h.com
Docdolittle	http://www.owlcentral.com
Docuscribe	http://www.docuscribe.com/index.html
eDrafter	www.docdev.com
FastDraft	http://www.fastdraft.com
GhostFill	http://www.ghostfill.com
Grantha	http://www.ssquaretech.com
HotDocs	http://www.capsoft.com
IntellX	http://www.business-integrity.com
IQDocs	http://www.iqdocs.com
KillerDocs	http://www.killerdocs.com
Lawgic	http://www.lawgic.com
Legal Ease	http://www.legal-ease.net
PowerTxt	http://www.interconweb.com/html/powertxt.html
ProDoc	http://www.prodoc.com
Rapidocs	http://www.rapidocs.com
SmartPrecedent	http://www.speedlegal.com/smartprecedent.html
SmartWords	http://www.lawontheweb.com
ThinkDocs	http://www.thinkdocs.com
Visual eForms	www.mmacorp.com
WinDraft	http://www.lawtech.com/WINDRAFT

Marc Lauritsen, president of Capstone Practice Systems (http://www.capstonepractice.com) practiced and supervised in legal aid offices for seven years, then served as an instructor, director of clinical programs, and a senior research associate at Harvard Law School. Marc directed Project PERICLES there, which focused on computer applications in legal services. He was chair of the American Bar Association's document assembly interest group and moderator of the law office automation forum on Counsel Connect. Last year he served as vice president for practice technology at AmeriCounsel.com, which developed an online environment for low-cost, high-quality legal service delivery through a nationwide network of private attorneys.